What is claimed is:

1. A method for providing an application based protocol suite, the method comprising:
   providing an active protocol stack to facilitate data communications across a data network; the active protocol stack including a network interface communicating directly to the data network to receive/send packets from/to the data network;
   providing an inactive protocol stack in the protocol suite, the inactive protocol stack comprising a number of components, each having characteristics suitable for one type of the data communications and being registered with a protocol manger; and
   activating one or more of the components from the inactive protocol stack into the active protocol stack when the one type of the data communications is detected.

2. The method of claim 1 further comprising:
   deactivating the one or more of the components from the active protocol stack back to the inactive protocol stack as soon as the one type of the data communications is done.

3. The method of claim 2 wherein the activating of the one or more of the components happens concurrently when one or more applications are executed to cause the one type of the data communications.

4. The method of claim 1 wherein the deactivating of the one or more of the components from the active protocol stack back to the inactive protocol stack is caused by the protocol manger.

5. The method of claim 4 wherein the protocol manger constantly monitors any new type of the data communications from the data network.

10092847.092601

6. The method of claim 1 wherein the network interface is a socket layer component and wherein the active protocol stack further includes a message handler component interfacing one or more applications based on the packets.

7. The method of claim 6 wherein the active protocol stack further includes a channel layer component responsible for routing/receiving the packets properly to/from each of the one or more applications.

8. The method of claim 7 wherein each of the components in the inactive protocol stack is a protocol component.

9. The method of claim 8 wherein the protocol component is a commonly used Internet protocol to support a particular data application.

10. The method of claim 9 wherein the protocol component is a user-defined protocol to support a user defined data type.

11. The method of claim 7 wherein each of the components in the inactive protocol stack is configured to have a common interface so that each of the components can be stacked one over another.

12. The method of claim 7 wherein each of the components in the inactive protocol stack and the network interface in the active protocol stack is configured to have a common interface so that each of the components can be stacked over the network interface when the each of the components is activated in the active protocol stack.

13. The method of claim 1 wherein the protocol manager keeps status of each of the components in the inactive protocol stack.

14. A network communications protocol program running on at least one of a network of computers for permitting and extracting communications from TCP/IP between hardware components including a plurality of processors connected over the computer network corresponding to one or more senders and a user of said communications, comprising:

a socket layer component for receiving one or more data packets from a first processor corresponding to a first sender via said network, and for defragmenting and reassembling data of said one or more data packets for demultiplexing and distributing first layer data portions of said one or more data packets into a plurality of channels according to which of a plurality of formats each of said first layer data portions resembles;

a channel layer component including said plurality of channels arranged according to which of said plurality of formats said data demultiplexed from said socket layer resembles, said channel layer for receiving said first layer data portions and demultiplexing and distributing second layer data portions from said first layer data portions according to which of a plurality of APIs each of said second layer portions is directed to; and

a message handler component including said plurality of APIs for receiving and handling said second layer data portions according to a plurality of user-defined responsibilities each of said APIs is configured to perform.

15. The program of Claim 14, said socket layer further for associating said one or more data packets with said first sender based on header data within said one or more data packets.

16. The program of Claim 14, said socket layer further for establishing a communications session between said user and said first sender, and sending a handshaking request response to said first processor.

09957369 .080601

17. The program of Claim 16, said socket layer further for associating said data packet with said first sender based on header data within said data packet.

18. The program of Claim 17, said socket layer further for exchanging generalized data specific to the first sender and the user.

19. The program of Claim 18, said generalized data including one or more of host URI, UUID and IP.

20. The program of Claim 19, said channel layer further for resolving said one or more of said host URI, UUID, and IP generalized data.

21. The program of Claim 18, said generalized data including prior communication history data between said first sender and user.

22. The program of Claim 14, said socket layer further for verifying and evaluating said data packet in a checksum subroutine.

23. The program of Claim 14, said channel layer further for prioritizing one or both of said first and second layer data portions based on a flexible hierarchy of priorities.

24. The program of Claim 23, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said user.

25. The program of Claim 24, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first sender.

26. The program of Claim 23, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first sender.

27. The program of Claim 14, said plurality of formats including two or more of voice, text, data, graphic, audio, video and combinations thereof.

28. The program of Claim 14, further comprising a decompression/decryption layer for decompressing and decrypting said one or more data packets depending on a state of compression and encryption of said data packets.

29. The program of Claim 14, further comprising a monitoring layer for tracking data exchanges between said user and said first sender.

30. The program of Claim 14, further comprising a data aggregation layer for aggregating data communicated by said first sender.

31. The program of Claim 14, further comprising a developer added layer for performing responsibilities defined by a developer.

32. The program of Claim 14, further comprising:

a decompression/decryption layer for decompressing and decrypting said one or more data packets depending on a state of compression and encryption of said data packets;

a monitoring layer for tracking data exchanges between said user and said first sender;

a data aggregation layer for aggregating data communicated by said first sender; and

a developer added layer for performing responsibilities defined by a developer.

33. A network communications protocol program running on at least one of a network of computers for permitting and extracting communications from TCP/IP between hardware components including a plurality of processors connected over the computer network corresponding to a user and one or more receivers of said communications, comprising:

a message handler component including a plurality of APIs for handling second layer data portions according to a plurality of user-defined responsibilities each of said APIs is configured to perform, and for configuring said second layer data portions for distribution to a plurality of channels according to which of a plurality of formats each of said second layer data portions resembles;

a channel layer component including a plurality of channels arranged according to which of said plurality of formats each of said second layer data portions resembles, said channel layer for multiplexing said second layer data portions into first layer data portions and channeling said first layer data portions according to which of said plurality of channels each of said first layer data portions is directed to; and

a socket layer component for receiving, multiplexing, disassembling and refragmenting said first layer data portions of each of said channels into one or more data packets, and communicating said one or more data packets over said network to one or more receivers corresponding to one or more processors over said network.

34. The program of Claim 33, said socket layer further for incorporating header data within said one or more data packets for associating said one or more data packets with said user.

35. The program of Claim 33, said socket layer further for establishing a communications session between said user and at least a first of said one or more receivers, and sending a handshaking request to a first processor corresponding to said first receiver.

36. The program of Claim 35, said socket layer further for incorporating header data within said one or more data packets for associating said one or more data packets with said user.

37. The program of Claim 36, said socket layer further for exchanging generalized data specific to the first receiver and the user.

38. The program of Claim 37, said generalized data including one or more of host URI, UUID and IP.

39. The program of Claim 38, said channel layer further for resolving said one or more of said host URI, UUID, and IP generalized data.

40. The program of Claim 37, said generalized data including prior communication history data between said first receiver and user.

41. The program of Claim 33, said channel layer further for prioritizing one or both of said first and second layer data portions based on a flexible hierarchy of priorities.

42. The program of Claim 41, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said user.

43. The program of Claim 42, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first receiver.

44. The program of Claim 41, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first receiver.

09923741.080601

45. The program of Claim 33, said plurality of formats including two or more of voice, text, data, graphic, audio, video and combinations thereof.

46. The program of Claim 33, said socket layer further for communicating said one or more data packets over said network to a first receiver corresponding to a first processor and a second receiver corresponding to a second processor.

47. The program of Claim 46, said socket layer further for incorporating header data within said one or more data packets for associating said one or more data packets with said user.

48. The program of Claim 46, said socket layer further for establishing a communications session between said user and said first and second receivers, and sending a handshaking request to said first and second processors.

49. The program of Claim 48, said socket layer further for incorporating header data with said one or more data packets for associating said one or more data packets with said user.

50. The program of Claim 49, said socket layer further for exchanging first and second generalized data specific to the first and second receivers, respectively, and the user.

51. The program of Claim 50, said generalized data including one or more of host URI, UUID and IP.

52. The program of Claim 51, said channel layer further for resolving said one or more of said host URI, UUID, and IP generalized data.

53. The program of Claim 51, said first and second generalized data including prior communication history data between said first and second recievers, respectively, and the user.

54. The program of Claim 46, said channel layer further for prioritizing one or both of said first and second layer data portions based on a flexible hierarchy of priorities.

55. The program of Claim 54, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said user.

56. The program of Claim 55, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first and second receivers.

57. The program of Claim 54, said channel layer further for updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first and second receivers.

58. The program of Claim 33, further comprising a compression/encryption layer for compressing and encrypting said one or more data packets depending on a user defined state of compression and encryption of said data packets.

59. The program of Claim 33, further comprising a monitoring layer for tracking data exchanges between said user and said recipients.

60. The program of Claim 33, further comprising a data aggregation layer for aggregating data communicated by said user.

09923771.080601

61. The program of Claim 33, further comprising a developer added layer for performing responsibilities defined by a developer.

62. The program of Claim 33, further comprising:

a compression/encryption layer for compressing and encrypting said one or more data packets depending on a user defined state of compression and encryption of said data packets;

a monitoring layer for tracking data exchanges between said user and said recipients;

a data aggregation layer for aggregating data communicated by said user; and

a developer added layer for performing responsibilities defined by a developer.

63. A method for permitting and extracting communications from TCP/IP between hardware components including a plurality of processors connected over the computer network corresponding to one or more senders and a user of said communications, comprising the operations:

receiving one or more data packets from a first processor corresponding to a first sender via said network;

defragmenting and reassembling data of said one or more data packets;

demultiplexing and distributing first layer data portions of said one or more data packets into a plurality of channels according to which of a plurality of formats each of said first layer data portions resembles;

further demultiplexing and distributing second layer data portions of said first layer data portions according to which of a plurality of APIs each of said second layer portions is directed to; and

handling said second layer data portions according to a plurality of user-defined responsibilities each of said APIs is configured to perform.

64. The method of Claim 63, further comprising the operation associating said one or more data packets with said first sender based on header data within said one or more data packets.

65. The method of Claim 63, further comprising the operations:

establishing a communications session between said user and said first sender; and

sending a handshaking request response to said first processor.

09523771.080601

66. The method of Claim 65, further comprising the operation associating said one or more data packets with said first sender based on header data within said one or more data packets.

67. The method of Claim 66, further comprising the operation exchanging generalized data specific to the first sender and the user.

68. The method of Claim 67, said generalized data including one or more of host URI, UUID and IP.

69. The method of Claim 68, further comprising the operation resolving said one or more of said host URI, UUID, and IP generalized data.

70. The method of Claim 67, said generalized data including prior communication history data between said first sender and user.

71. The method of Claim 63, further comprising the operation verifying and evaluating said data packet in a checksum subroutine.

72. The method of Claim 63, further comprising the operation prioritizing one or both of said first and second layer data portions based on a flexible hierarchy of priorities.

73. The method of Claim 72, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said user.

74. The method of Claim 73, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first sender.

75. The method of Claim 72, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first sender.

76. The method of Claim 63, said plurality of formats including two or more of voice, text, data, graphic, audio, video and combinations thereof.

77. The method of Claim 63, further comprising the operation decompressing/decrypting said one or more data packets depending on a state of compression and encryption of said data packets.

78. The method of Claim 63, further comprising the operation monitoring data exchanges between said user and said first sender.

79. The method of Claim 63, further comprising the operation aggregating data communicated by said first sender.

80. The method of Claim 63, further comprising the operation performing responsibilities defined by a developer.

81. The method of Claim 63, further comprising the operations:

decompressing/decrypting said one or more data packets depending on a state of compression and encryption of said data packets;

monitoring data exchanges between said user and said first sender;

10980617.032502

aggregating data communicated by said first sender; and

performing responsibilities defined by a developer.

82. A method for permitting and extracting communications from TCP/IP between hardware components including a plurality of processors connected over the computer network corresponding to a user and one or more receivers of said communications, comprising the operations:

handling second layer data portions according to a plurality of user-defined responsibilities each of one or more APIs is configured to perform;

configuring said second layer data portions for distribution to a plurality of channels according to which of a plurality of formats each of said second layer data portions resembles;

multiplexing said second layer data portions into first layer data portions;

channeling said first layer data portions according to which of said plurality of channels each of said first layer data portions is directed to;

receiving, multiplexing, disassembling and refragmenting said first layer data portions of each of said channels into one or more data packets; and

communicating said one or more data packets over said network to one or more receivers corresponding to one or more processors over said network.

83. The method of Claim 82, further comprising the operation incorporating header data within said one or more data packets for associating said one or more data packets with said user.

84. The method of Claim 82, further comprising the operations:

establishing a communications session between said user and at least a first of said one or more receivers; and

sending a handshaking request to a first processor corresponding to said first receiver.

10980774.062501

85. The method of Claim 84, further comprising the operation incorporating header data within said one or more data packets for associating said one or more data packets with said user.

86. The method of Claim 85, further comprising the operation exchanging generalized data specific to the first receiver and the user.

87. The method of Claim 86, said generalized data including one or more of host URI, UUID and IP.

88. The method of Claim 87, further comprising the operation resolving said one or more of said host URI, UUID, and IP generalized data.

89. The method of Claim 86, said generalized data including prior communication history data between said first receiver and user.

90. The method of Claim 82, further comprising the operation prioritizing one or both of said first and second layer data portions based on a flexible hierarchy of priorities.

91. The method of Claim 90, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said user.

92. The method of Claim 91, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first receiver.

93. The method of Claim 90, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first receiver.

94. The method of Claim 82, said plurality of formats including two or more of voice, text, data, graphic, audio, video and combinations thereof.

95. The method of Claim 82, further comprising the operation communicating said one or more data packets over said network to a first receiver corresponding to a first processor and a second receiver corresponding to a second processor.

96. The method of Claim 95, further comprising the operation incorporating header data within said one or more data packets for associating said one or more data packets with said user.

97. The method of Claim 95, further comprising the operations:

establishing a communications session between said user and said first and second receivers; and

sending a handshaking request to said first and second processors.

98. The method of Claim 97, further comprising the operation incorporating header data with said one or more data packets for associating said one or more data packets with said user.

99. The method of Claim 98, further comprising the operation exchanging first and second generalized data specific to the first and second receivers, respectively, and the user.

100. The method of Claim 99, said generalized data including one or more of host URI, UUID and IP.

101. The method of Claim 100, further comprising the operation resolving said one or more of said host URI, UUID, and IP generalized data.

102. The method of Claim 99, said first and second generalized data including prior communication history data between said first and second recievers, respectively, and the user.

103. The method of Claim 95, further comprising the operation prioritizing one or both of said first and second layer data portions based on a flexible hierarchy of priorities.

104. The method of Claim 103, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said user.

105. The method of Claim 104, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first and second receivers.

106. The method of Claim 103, further comprising the operation updating said flexible hierarchy of priorities as said priorities evolve based on status and activities of said first and second receivers.

107. The method of Claim 82, further comprising the operation compressing/encrypting said one or more data packets depending on a user defined state of compression and encryption of said data packets.

108. The method of Claim 82, further comprising the operation monitoring data exchanges between said user and said recipients.

109. The method of Claim 82, further comprising the operation aggregating data communicated by said user.

110. The method of Claim 82, further comprising the operation performing responsibilities defined by a developer.

10030712 "101301

111. The method of Claim 82, further comprising the operations:

compressing/encrypting said one or more data packets depending on a user defined state of compression and encryption of said data packets;

monitoring data exchanges between said user and said recipients;

aggregating data communicated by said user; and

performing responsibilities defined by a developer.